MULTIMEDIA UNIVERSITY

STUDENT ID NO

MULTIMEDIA UNIVERSITY

FINAL EXAMINATION

TRIMESTER 1, 2015/2016

DCS5088 – OBJECT ORIENTED PROGRAMMING

(For DIT Students Only)

15 OCTOBER 2015
9.00 a.m. – 11.00 a.m.
(2 Hours)

INSTRUCTIONS TO STUDENTS

1. This examination paper consists of **14** pages of questions excluding the cover page. There are THREE (3) sections in this paper.

2. **SECTION A:** There are **TEN (10)** TRUE/FALSE questions. Answer **ALL** questions on the answer booklet provided (10 marks).

3. **SECTION B:** There are **TWO (2)** structured questions. Answer **ALL** questions on the answer booklet provided (40 marks).

4. **SECTION C:** There is **ONE (1)** structured question. Answer the question on the answer booklet provided (30 marks).

SECTION A [10 MARKS]

*Instruction: Answer **'True'** or **'False'** for all statements below and write your answers on the answer booklet provided.*

1. Encapsulation hides the details of the process from the user.

2. Linker is a program that combines the object code that the compiler produced with the object codes that are already programmed (in a library).

3. There are 3 types of errors or mistakes that cause the program to behave unexpectedly; syntax errors, run-time errors, and source code errors.

4. The six phases involve in translating a C++ program into an executable file are; edit, preprocess, compile, link, load, execute.

5. The *break* statement is required in the *default* case of a *switch* selection statement to exit the switch properly.

6. An address operator (&) is one that can hold address of a variable to indirectly refer to another variable.

7. *while* loops, *do-while* loops and *for* loops are the common control structures used for performing iteration statements in C++ program.

8. A class is an implementation of an Abstract Data Type (ADT).

9. The concept of object oriented is implemented in C++ using classes. One of the similarities between a class and a struct is permitting the definition of data and operations / functions to be performed.

10. The methods declared in a class are called data members.

Continued.....

SECTION B **[40 MARKS]**

Instruction: Answer all questions and write your answers on the answer booklet provided. You do not have to write a full program for each of the questions below.

QUESTION 1 (20 marks)

1.1 Complete the calorie-counting program at the segments labelled :

(a) *'//1.1 (a)'* – Complete the body of the loop by writing the codes to get input for *calories_for_item* and to determine the *total_calories*. Then, display the *total_calories* amount. [3.5 marks]

(b) *'//1.1 (b)'* – Based on the table below, write complete nested *if-else if* statements to output the correct message. [3.5 marks]

Gender	*Total calories required per day*	*Output message*
F	1600 to 2400	For a female, you're eating healthily !
	Less than 1600 or more than 2400	For a female, this is unhealthy eating !
M	2400 to 3000	For a male, you're eating healthily !
	Less than 2400 or more than 3000	For a male, this is unhealthy eating !
Other than the above	-	Invalid gender input

```
#include <iostream>
using namespace std;

int main( )
{
    int number_of_items, count,
        calories_for_item, total_calories;
    char gender;

    cout<< "Enter F for female and M for male: \n";
    cin >> gender;

    cout << "How many items did you eat today? ";
    cin >> number_of_items;

    total_calories = 0;
    count = 1;
    cout << "Enter the number of calories in each of the\n"
         << number_of_items << " items eaten:\n";

    while (count++ <= number_of_items)
```

Continued.....

```
    // 1.1 (a) Write your answer on your answer booklet

    // 1.1 (b) Write your answer on your answer booklet

    return 0;
}
```

[**Note:** Refer to sample input/output given below. The bold and italic items are inputs to the program by user]

Sample input/output screen

```
Enter F for female and M for male:
F

How many items did you eat today? 5

Enter the number of calories in each of the 5 items eaten:

200 1500 1400 230 10

Total calories eaten today = 3340

For a female, this is unhealthy eating !
```

1.2 Suppose that your program contains the following class definition:

```
class Seminar
{
  public:
       void set_name(string n);
       string get_name();
  private:
       string name;

};
```

and suppose the ***main()*** of your program contains the following declaration:

```
Seminar T1, T2, *P;
```

State whether the following statements are **valid** or **invalid** in the *main()*. If invalid, provide the reason why it is **invalid**. [7 marks]

i. `T1.name = "Google apps for Education";`
ii. `T2.set_name("Customer Service Excellence");`
iii. `P = T1;`
iv. `cout << "Name :" << get_name() <<endl;`

Continued.....

1.3 Given the following incomplete program, answer the following questions:

```
#include <iostream>
#include <string>
using namespace std;
class Product
{   int rate[10];
    string id;

    public:
      void setID();
      void setRate()
      { cout <<"Enter 10 rating values (range 1 - 5) "
                  "for this product : \n";

         for(int i=0; i<10; i++)
         {  cin>>rate[i];
            while(rate[i] < 0 || rate[i] > 5)
           { cout << "Invalid value, please enter a new value\n";
             cin >>rate[i];
           }
         }
      }

      void outputChart()
      {   int level[5] = {0};
          cout << "\nProduct ID " <<id<<" rating Chart...." <<endl;
          for(int i=0; i<10; i++)
           {  if(rate[i] == 1) level[0]++;
              else if(rate[i] == 2) level[1]++;
              else if(rate[i] == 3) level[2]++;
              else if(rate[i] == 4) level[3]++;
              else level[4]++;
           }

          // 1.3 (a) Write your answer on your answer booklet

      }

};

// 1.3 (b) Write your answer on your answer booklet

int main( )
{
      Product Bear;
      Bear.setID();
      Bear.setRate();
      Bear.outputChart();
      return 0;

}
```

Continued.....

Sample input/output screen

```
Enter product ID : AC201
Enter 10 rating values (range 1 - 5) for
this product :
3
3
20
Invalid value, please enter a new value
25
Invalid value, please enter a new value
2
2
3
1
5
4
4
4

Product ID AC201 rating Chart....
Level 1: *
Level 2: **
Level 3: ***
Level 4: ***
Level 5: *
```

[**Note:** Refer to sample input/output given above. The bold and italic items are inputs to the program by user]

(a) At the segment labelled *'//1.3(a)'*:

- Write the codes to display the bar chart showing the asterisks (*) based on the frequency of the levels. [4 marks]

(b) At the segment labelled *'//1.3(b)'*:

- Write the codes that define member function ***void setID()*** **outside of the class.** Get user input for ***id***. [2 marks]

Continued.....

QUESTION 2 (20 marks)

2.1 Given the following program, answer the following questions:

```
#include <iostream>
using namespace std;
class Number
{     int x, y;
      public:
            int findresult()
            { int tmp, x1, y1;
              x1 = x;
              y1 = y;
              while (y1!= 0)
              {
                tmp = x1;
                x1 = y1;
                y1 = tmp % y1;
              }
              return x1;
            }

            Number(int a, int b)
            {  x = a;
               y = b;
            }

            int getX()
            { return x; }

            int getY()
            { return y; }
};

int main()
{
  Number NR(3,12);

  cout << "Given " << NR.getX() << " and " << NR.getY() <<
  ", the number you are looking for is: "<< NR.findresult() <<endl;

  return 0;
}
```

(a) Trace and write the output produced by the program. Show your workings or diagrams to support your answer. [4 marks]

(b) Change ***int findresult()*** to become a ***friend*** function of the *Number* class. What are the necessary changes that you will make? Illustrate the updates

Continued.....

you will make to the program above. (Note: You do not need to rewrite the entire program but explain briefly **where** and **what** changes that need to be made) [4 marks]

2.2 Given the complete program below, answer the following questions:

```
#include <iostream>
using namespace std;

class Customer
{
     int cust_id; // error

};

class Member : public Customer
{
      int points, cumulative;

      public:
      Member(int pt, int id)
      {  points = pt;
         cumulative = pt;
         cust_id = id ;
      }

      // 2.2 (c) Write your answer on your answer booklet


     void display_points(int p)
     {  cout << "Cust ID\t\t: " << cust_id << endl;
        if  ( p == 0 )
           cout << "Points\t\t: No points this time " <<endl;
        else
        {  if ( p == -1 )
              cout << "Current Points\t: " << points << endl;
           else
           { points = p ;
             cout <<"Points\t\t: " << points << endl;
           }
         }

        cout << "Cumulative\t: " << cumulative << endl << endl;
        points = cumulative;
      }
};

class Purchase
{
   float amount;
```

Continued.....

```
   string receipt;

   public:
        Purchase(float m, string r)
        {  amount = m;
           receipt = r;
        }

         int findpoints()
         {  int points;
            cout <<"--Receipt for this purchase is " << receipt
            <<endl;
            if (amount < 50)
            {    cout << "No points" <<endl;
                 points = 0;
            }
            else
                 points = amount*2;
            return points;
         }

};

int main()
{  Member M2(300, 1116);
   Purchase A(100.00, "A111");
   Purchase B(49.00, "AB121");
   int points;

   M2.display_points(-1);
   points = A.findpoints();
   M2+=points;
   M2.display_points(points);
   points = B.findpoints();
   M2+=points;
   M2.display_points(points);
   return 0;
}
```

Sample input/output screen

```
Cust ID          : 1116
Current Points   : 300
Cumulative       : 300

--Receipt for this purchase is A111
Cust ID          : 1116
Points           : 200
Cumulative       : 500

--Receipt for this purchase is AB121
No points
Cust ID          : 1116
```

Continued.....

```
Points          : No points this time
Cumulative      : 500
```

(a) Identify the derived class. [1 mark]

(b) There is an error in the program above (highlighted and label '//error'). Explain why it is an error and how to resolve it. [2 marks]

(c) At the segment labelled *'//2.2 (c)'*, define overloaded operator '+=' member function. The function has one integer parameter. Increment the data member ***cumulative*** with this parameter value. [3 marks]

2.3 Trace and write the output produced by the program below: [6 marks]

```
#include<iostream>
using namespace std;

class Greeting
{    protected:
        string message;

     public:
       Greeting()
       { message = "Happy Birthday...";
       }

         virtual void display ( string n )
         {  cout<< message <<", "<< n << endl;
         }

         virtual ~Greeting()
         { cout<< "Greeting class" << endl << endl; }
};

class Special : public Greeting
{   int set;

    public:
      Special()
      { set = 1;
      }

      void choice(int n)
      { set = n;
        if(set==1)
              message = "Happy Birthday to my dear sister";
        else if (set==2)
             message = "Happy Birthday to my dear brother";
```

Continued.....

```
      }

      void display( string n )
      {  static int num=1;
         cout<< message <<", "<< n << endl;
         add();
         cout <<"That was message no:" <<num<<endl<<endl;
         num++;
       }

       void add()
       { cout<< "You're simple amazing" <<endl;
       }
};

int main()
{   Greeting j;
    Special s1, s2;

    Greeting *p = &s1;
    s1.choice(2);
    p->display("Tim");

    p = &s2;
    s2.choice(1);
    s2.display("Maria");


    p = &j;
    p->display("Ann-Marie");

   return 0;
}
```

Continued.....

SECTION C **[30 MARKS]**

Instruction:
Write a **complete program** that determines the eligibility result of a loan applicant.

Create a class called *Dependent :*

- Public data members:
  - *name* : string
  - *relationship* : string

Create a class called *LoanApplicant* :

- Private data members:
  - *name* : string
  - *icno* : icno
  - *no_of_dep* : int
  - *d* : pointer object of Dependent class
  - *amount* : float
  - *yearly_salary* : float

- Public member functions:
  - *setDetails ( )* : - Get user inputs for *name, icno, yearly_salary, amount* and *no_of_dep.*
    - Create a dynamic array of *Dependent* objects using pointer *d*, based on the size which is *no_of_dep.*
    - Using a *for* loop, get user input for the name and relationship for every *Dependent* object element of the array.
  - *check( )* :- Determine the eligibility of the loan applicant using nested if-else if statements.

Criteria	*Other criterias*	*Display message*
Amount greater than or equal to 10,000.	-	"Sorry, this amount is not catered through the application"
Amount lesser than 10,000.	No of dependents less than or equals to 1 and yearly salary greater than or equals to 60,000.	"You're only eligible for a loan of 1/2 amount of the original which is RM ...." *Note: The new*

Continued.....

		amount is calculated and displayed in the message
	No of dependents greater than or equals to 2 and yearly salary less than 60,000.	"You're eligible for full loan amount which is RM ..." *Note: The full amount applied for is displayed in the message.*
	-	"Please visit our Loan Officer to check on other criterias."

- *display( )* : - Display *name*, *icno* and *no_of_dep*.
  - Using a *for* loop, display the name and relationship for every *Dependent* object element of the array.
  - Call *check()*.

- *Destructor function* :- Deallocate the memory of the dynamic array of *Dependent* objects.

In *main( )*:

- Declare necessary variables depending on the requirements.
- Using a **do-while** sentinel controlled loop :
  - Declare a *LoanApplicant* object:
    - Call *setDetails()* and *display()*
  - Keep track of how many times the loop has executed.
  - Prompt the user whether to continue. If user indicates 'yes' (enters 'Y' or 'y'), continue looping.
- Display the total number of applicants (the number of times the loop has executed).

[**Note:** Refer to sample input/output given below. The bold and italic items are inputs to the program by user]

Continued.....

Sample input/output screen

```
------------------------------------------------------------
                    INPUT DETAILS
------------------------------------------------------------
Enter name                : Mandy Moore
Enter IC no               : 781005-04-5144
Enter yearly salary       : RM 55000
Enter loan amount         : RM 3500
Enter no of dependents    : 1
---------------Set Dependent Information--------------------
Enter name              : James Lee
Enter relationship      : Son

------------------------------------------------------------
                    DISPLAY DETAILS
------------------------------------------------------------
Name              : Mandy Moore
IC NO             : 781005-04-5144
No of Dependents :1
-------------------Dependent Information--------------------
Dependent [1]  and relationship    : James Lee , Son

---Loan Result----- >>>>Please visit our Loan Officer to check on
other criterias.


Continue? [Y/N] :Y

------------------------------------------------------------
                    INPUT DETAILS
------------------------------------------------------------
Enter name                : Andy Ng
Enter IC no               : 790605-01-6781
Enter yearly salary       : RM 61000
Enter loan amount         : RM 11000
Enter no of dependents    : 2
---------------Set Dependent Information--------------------
Enter name              : Karina Ng
Enter relationship      : Daughter
---------------Set Dependent Information--------------------
Enter name              : Maria Ang
Enter relationship      : Wife


------------------------------------------------------------
                    DISPLAY DETAILS
------------------------------------------------------------
Name              : Andy Ng
IC NO             : 790605-01-6781
No of Dependents :2
-------------------Dependent Information--------------------
Dependent [1]  and relationship    : Karina Ng , Daughter
Dependent [2]  and relationship    : Maria Ang , Wife

---Loan Result----- >>>>Sorry, this amount is not catered through
the application.

Continue? [Y/N] :Y
------------------------------------------------------------
```

Continued.....

```
                    INPUT DETAILS
-------------------------------------------------------------
Enter name                : Vera Wang
Enter IC no               : 770609-04-5178
Enter yearly salary       : RM 65000
Enter loan amount         : RM 5000
Enter no of dependents    : 1
---------------Set Dependent Information--------------------
Enter name              : Aireen Lim
Enter relationship      : Daughter

-------------------------------------------------------------
                    DISPLAY DETAILS
-------------------------------------------------------------
Name              : Vera Wang
IC NO             : 770609-04-5178
No of Dependents :1
-------------------Dependent Information--------------------
Dependent [1]  and relationship    : Aireen Lim , Daughter

---Loan Result-----  >>>>You're only eligible for a loan of 1/2
amount of the original which is RM 2500.

Continue? [Y/N] :Y

-------------------------------------------------------------
                    INPUT DETAILS
-------------------------------------------------------------
Enter name                : Yew Chi Keong
Enter IC no               : 680104-04-9881
Enter yearly salary       : RM 50000
Enter loan amount         : RM 2300
Enter no of dependents    : 2
---------------Set Dependent Information--------------------
Enter name              : Victor Yew
Enter relationship      : Son
---------------Set Dependent Information--------------------
Enter name              : Angeline Ng
Enter relationship      : Wife

-------------------------------------------------------------
                    DISPLAY DETAILS
-------------------------------------------------------------
Name              : Yew Chi Keong
IC NO             : 680104-04-9881
No of Dependents :2
-------------------Dependent Information--------------------
Dependent [1]  and relationship    : Victor Yew , Son
Dependent [2]  and relationship    : Angeline Ng , Wife

---Loan Result-----  >>>>You're eligible for full loan amount which
is RM 2300.

Continue? [Y/N] :N


::Total number of Applicants :4
```

End of Page.